

데이터 통신(02 반)

10 주차 실습/과제 보고서



학번 201902684

학과 컴퓨터융합학부

이름 노아영

<과제 목표>

- python 의 socket 모듈을 사용한 파일 전송 프로그램 만들기
- 윈도우와 가상머신 사이의 통신

<해결 방법>

①Sender(window)

1. 인자값 확인

1-a) len()을 사용한 인자 개수 확인 → 틀릴 경우 sys.exit()사용

1-b) python3 sender.py [port_number] → 변수에 sys.argv[1]을 통해 값 할당

⇒ port = sys.argv[1]

```
if len(sys.argv) != 2:
    print("Fail! -- Require two arguments.")
    sys.exit()
port = sys.argv[1]
```

2. 소켓 생성(UDP 소켓 생성) 및 자료 수신 준비

```
try:
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    sock.bind(("192.168.56.1", int(port)))
except socket.error:
    print("failed to create socket")
    sys.exit()
```

⇒ 윈도우 ip 주소인 192.168.56.1 와 포트번호 8000(인자값)을 사용

3. receiver 가 보내는 명령을 확인하여 진행

: receiver 가 보내는 명령(data) 확인 후 만약 exit 일 경우 socket.close()와 sys.exit()을 진행하고 receive [파일 이름]일 경우 sender_send() 함수를 실행한다.

```
try:
    data, addr = sock.recvfrom(2048)
    data = data.decode()
except ConnectionResetError:
    print("error. port number not matching.")
    sys.exit()

handler = data.split()

if handler[0] == 'receive':
    sender_send(handler[1])
else:
    print("exit")
    sock.close()
    sys.exit()
```

4. sender_send() 내부 구현

4-a) receiver 쪽으로 receive 명령어를 잘 받았다고 알림을 보냄

4-b) 파일이 존재하는지 확인하고 receiver 쪽으로 알림을 보냄

4-c) 전달받은 파일의 크기를 확인하고 receiver 쪽으로 읽기 진행 횟수(n)를 보냄

4-d) 파일을 바이트 형태로 4-c)에서 구한 크기만큼의 내용을 receiver 에 보냄

```
def sender_send(file_name):
    # 4-a
    sock.sendto("valid list command.".encode('utf-8'), addr)
    # 4-b
    if os.path.isfile(file_name):
        # 4-c
        size = os.stat(file_name).st_size
        n, cnt = int(size/2048), 0
        sock.sendto("file exists!".encode('utf-8'), addr)
        sock.sendto(str(n).encode('utf-8'), addr)

        read_file = open(file_name, "rb") # read for bytes
        print("Start sending file.")

        # 4-d
        while n:
            r = read_file.read(2048)
            sock.sendto(r, addr)
            cnt += 1
            print("packet number:", cnt)
            n -= 1
        print("Finished sending file.")
    else:
        sock.sendto("file not exists!".encode('utf-8'), addr)
```

⇒ 각 상황에 맞는 문장을 print()!

②Receiver(virtual box)

1. 인자값 확인

1-a) len()을 사용한 인자 개수 확인 → 틀릴 경우 sys.exit()사용

1-b) python3 sender.py [port_number] → 변수에 sys.argv[1]을 통해 값 할당

⇒ ip_address = sys.argv[1], port_number = sys.argv[2]

```
if len(sys.argv) != 3:
    sys.exit()
ip_address = sys.argv[1]
port_number = sys.argv[2]
```

2. 소켓 생성(UDP 소켓 생성) 및 non-blocking 설정, timeout 설정

```
try:
    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    s.setblocking(0)
    s.settimeout(15)
    print("receiver socket created")
except socket.error:
    print("failed to create socket")
    sys.exit()
```

⇒ 소켓 생성: s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

⇒ Non-blocking 설정: s.setblocking(0)

⇒ Timeout 설정: s.settimeout(15) # timeout 15 로 설정

3. 소켓 생성(UDP 소켓 생성) 및 non-blocking 설정, timeout 설정

```
read = input("enter a command: \n1.receive [file_name]\n2.exit\n")
#
# send to sender
# "receive " + file_name
#
s.sendto(read.encode(), (ip_address, int(port_number)))

if read == "exit":
    sys.exit()
```

⇒ input()으로 명령어를 입력 받은 후 명령어를 sender 쪽으로 보냄

⇒ 명령어가 exit 일 경우 sys.exit()으로 빠져나옴, receive [파일 이름]일 경우 코드를 이어서 진행함

4. 파일 받기

4-a-i) valid: sender 가 명령어를 정상적으로 받았는지 확인

4-a-ii) exist: sender 쪽에 파일이 존재하는지 확인

4-a-iv) n: 파일을 몇 개로 나누어 전송하는지 전달받기

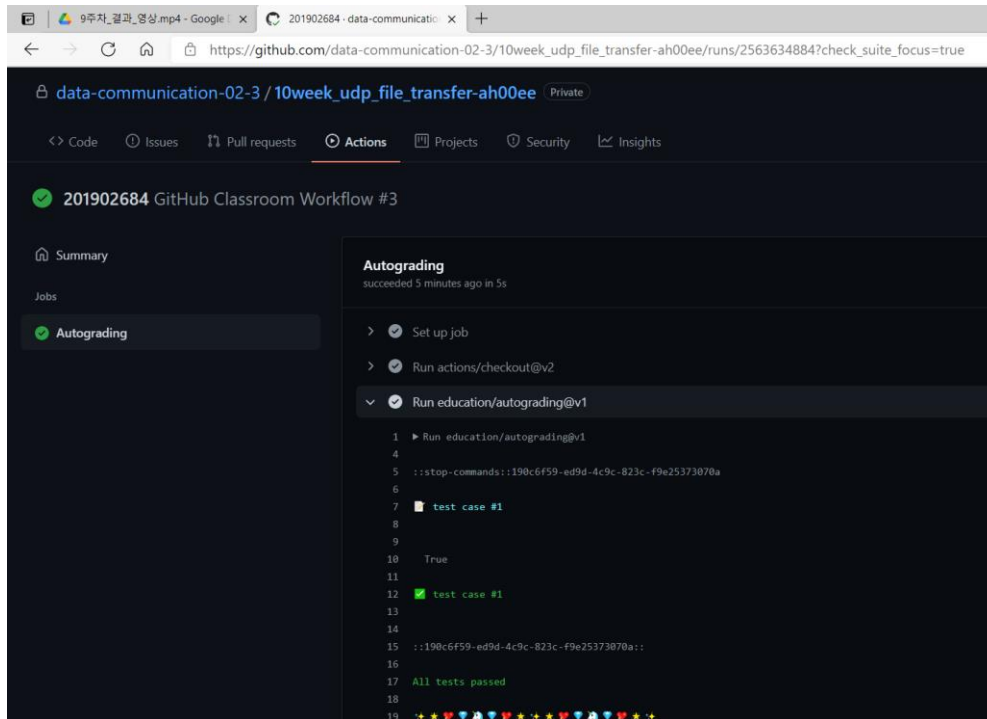
```
valid = s.recvfrom(2048)[0]
print(valid.decode('utf-8'))
#
# if receiver exist msg:
#
exist = s.recvfrom(2048)[0]
n = int(s.recvfrom(2048)[0].decode('utf-8'))
print(exist.decode('utf-8'))
```

4-a-iii) & 4-a-v) 파일을 전송받고 저장함

```
f = open("received_file.txt", "wb")
print("Start receiving file.")
cnt = 0
while n:
    data = s.recvfrom(2048)[0]
    f.write(data)
    cnt += 1
    print("Received packet number:", cnt)
    n -= 1
#
# file_name.close()
#
f.close()
print("Finished receiving file. Check file in directroy.")
```

- ⇒ n 횟수만큼 파일을 전송받고 전송 상황을 print()
- ⇒ while 문을 다 돌면 파일을 닫음(=f.close())
- ⇒ 파일을 다 받았다고 안내

<Github Classroom 통과>



<Youtube 링크>

<https://youtu.be/xm37XZ37dsw>

** 영상 마지막에 ls 시 나오는 파일에서 파일 생성이 되지 않은 것이 아닌 같은 파일명으로 파일을 생성하여 먼저 불러온 steve_jobs_speech_script.txt 파일에 덮어쓰기 된 것입니다!